基于多项式和混沌序列的无线传感器网络密钥管理方案 *

胡聚宁, 毕红军, 刘 云, 贾 凡
(北京交通大学 通信与信息系统北京市重点实验室, 北京 100044)

摘 要: 针对现有密钥管理方案占据存储空间大和安全性差的缺点,提出了一种基于多项式和混沌序列的混合密钥管理方案。该方案使用多项式建立节点间的共享密钥减少网络的存储开销,利用混沌序列对初值的敏感性特点解决多项式安全门限的问题,从而防止密钥的泄露。结果表明,本方案具有很高的安全性能,不仅节约了网络的通信资源,而且降低了节点的存储开销。

关键词: 无线传感器网络; 密钥管理; 多项式; 混沌序列; 安全门限
中图分类号: TP393 **文献标志码**: A **文章编号**: 1001-3695(2010)07-2602-03
doi:10.3969/j.issn.1001-3695.2010.07.056

Key management scheme based on polynomial and chaos for wireless sensor networks

HU Ju-ning, BI Hong-jun, LIU Yun, JIA Fan
(*Key Laboratory of Communication & Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing* 100044, *China*)

Abstract: Due to the large storage and poor security of current schemes, this paper presented a hybrid key management scheme based on polynomial and chaotic sequence. The scheme reduced the network storage through polynomial to establish a shared key between nodes, and solved safe threshold problem in polynomial scheme by taking the advantage of the sensitivity of chaos on initial value, preventing the leakage of keys. The results indicate that the scheme can provide a high level of security performance, not only saves the network communication resources, but also loweres the storage of nodes.
Key words: wireless sensor networks(WSN); key management; polynomial; chaos; secure threshold

0 引言

无线传感器网络(WSN)是由部署在监测区域内大量的微型传感器节点组成的,通过无线通信方式形成的一个多跳的自组织网络系统,其目的是协作地感知、采集和处理信息,把最终数据发送给用户。由于 WSN 具有无可比拟的优势,它被广泛应用于军事国防、工农业控制、危险区域远程控制等领域[1]。若 WSN 部署在敌方环境时,节点将面临各种各样的攻击,因此如何保证网络的安全性是近年来研究的热点。然而无线传感网络在能量消耗、计算能力、通信带宽等方面受到严格的限制,传统的基于公钥和可信第三方的密钥分配协议并不适用于 WSN。

基于以上原因,研究者们提出了多种密钥管理方案来满足 WSN 的要求[2]。Eschenauer 等人[3]从安全性和节点存储空间考虑,提出了基于概率论的随机密钥预分配方案。该方案的基本思想是节点从密钥池中随机选取一部分密钥作为自己的密钥环,网络预分配之后,只要两个节点有一对相同的密钥就可以直接建立安全的通信,它的缺陷在于密钥池的大小决定了网络的连通性和安全性。Chan 等人[4]在基本随机密钥预分配的基础上提出了 q-composite 的密钥对预分配方案,其要求两个节点之间至少需要 q 个相同的密钥才能直接建立共享链路密钥。这种方案在一定程度上提高了网络的安全性,但是当被捕获节点超过某个范围时,网络的安全性下降得更快。Liu Dong-gang 等人[5]在基于概率论方案和多项式方案的基础上,用多项式池代替密钥池,在多项式构造中应用安全门限值较大地提高了安全性,但同时也增加了计算负载。之后,Du Wen-liang 等人[6,7]提出了多重空间的密钥预分配方案,提高了密钥环中密钥的利用率。Traynor 等人[8]提出了适用于异构传感器网络的密钥预分配方案。在该方案中,节点按照存储能力、计算能力等被分为 H-节点和 L-节点,发挥 H-节点能量充足、通信范围大、存储能力强的作用,降低了网络的能量消耗,同时达到了比较理想的网络安全性。但是当大量 H-节点遭到破坏时,网络的安全性将受到致命的威胁。

针对已有的层簇式传感器网络结构,本文借助于多项式并结合混沌序列的遍历性和对初值的敏感性,提出了一种混合密钥预分配方案。

收稿日期: 2009-12-16; **修回日期**: 2010-02-13 **基金项目**: 国家自然科学基金资助项目(60972012);国家“863”计划资助项目(2009AA01Z423);北京市重点实验室基金资助项目;北京市教育委员会学科建设与研究生建设资助项目(JXKJD20090001)

作者简介:胡聚宁(1986-),男,河北邢台人,硕士,主要研究方向为网络安全、无线传感器网络等(clarm1215@126.com);毕红军(1965-),男,副教授,博士,主要研究方向为密码学、网络安全;刘云(1955-),女,教授,博士,主要研究方向为信息网络、网络安全、智能交通;贾凡(1976-),男,讲师,博士,主要研究方向为网络安全.

1　基于多项式的密钥管理

Blundo 等人[9]在随机密钥对的基础上提出了一种有限域 $F(Q)$ 上的 t 阶多项式的密钥预分配方案。在此方案中,系统随机生成一个 t 阶二元多项式:

$$f(x,y)=\sum_{i,j=0}^{t}a_{ij}x^iy^j \tag{1}$$

其中:Q 是与密钥长度相适应的大素数;系数 a_{ij} 为有限域内选取的随机数。从式(1)可以看出,二元多项式具有对称的特性,即 $f(x,y)=f(y,x)$。

在节点部署之前,系统为每个节点预分配多项式。对于任意的节点 u,系统预分配多项式 $f_u(y)=f(u,y)$;相应地,对于节点 v,系统预分配多项式 $f_v(y)=f(v,y)$。在网络部署之后,节点广播自己的密钥信息,在收到邻居节点的信息后,节点 u 计算 $f(u,y)$在 $y=v$ 的值,节点 v 计算 $f(v,y)$在 $y=u$ 的值,根据多项式的对称性,节点 u 和 v 即可建立共享密钥 k_{uv},即 $f(u,v)=f(v,u)$。这种方案是用多项式代替密钥池,兼顾节点存储容量和网络安全连通的特点,一定程度上提高了网络的安全性,但是它存在安全门限 t 的问题。也就是说,当被捕节点的数量不超过 t 时,被捕节点不会泄露其他节点的密钥信息;但是当被捕节点的数量大于 t 时,攻击者可以重构多项式而使网络的安全性迅速下降。

2　混沌序列

混沌[10]是非线性系统中存在的一种普遍现象,是指连续或者离散动力系统产生的一种对初始条件具有敏感依赖性的回复性非周期运动。混沌最重要的特征是对初值的敏感性,也就是说,初值的微小变化都会引起运算结果的巨大差别;同时,整个序列的遍历性和非周期性形成了系统不可预测的行动,形成了混沌的特殊序列。在基于混沌的无线传感器网络密钥预分配方案中,采用 Logistic 映射进行密钥预分配,利用混沌的遍历性和初值敏感性来提高系统的安全性。

Logistics 映射是一个典型的非线性混沌方程,其定义为

$$f(x_0,\lambda,n):x_{n+1}=\lambda x_n(1-x_n) \tag{2}$$

其中:$x_n\in(0,1)$ 为系统的状态变量,$n=1,2,3,\cdots$;$\lambda\in(0,4)$ 为系统的参数,当 $\lambda<3.0$ 时,函数趋于稳定,当 $\lambda>3.0$ 时,函数开始出现分叉现象,当 $3.5699456\cdots<\lambda\leqslant4$,函数产生混沌序列。

因此,混沌序列可以应用到产生密钥的过程中。对任意的节点,只需存储相同的密钥,即函数的初始状态参量$\langle x_0=x_s,\lambda_0=\lambda_s\rangle$。当节点 u 与节点 v 通信时,节点 u 随机产生一个正整数 R_{uv},然后计算 $k_{u*}=\phi(f(x_0,\lambda_0,n))$在 $n=R_{uv}$的值。其中映射函数 $k_i=\phi(x_i)$将选择 k_i 小数点后若干位(依网络需求而定),并将其组合成一个整数 k_{uv}。然后,节点 u 通过一定的方式将随机数 R_{uv}传给节点 v,节点 v 通过同样的方法计算出 k_{vu},$k_{uv}=k_{vu}$即为两者通信的共享密钥。

3　基于多项式和混沌序列的无线传感器网络密钥管理方案

3.1　前提条件和假设

本方案是基于以下假设的:a)基站或者可信的第三方是传感器网络中可信任且无法被捕获的节点;b)每一个传感器节点在部署之前都是完好的,并且在部署之后很短的一段时间内不会受损,因为捕获一个节点比节点间建立密钥对花费更长的时间。该方案包括密钥预分配、密钥的建立和密钥的动态管理三部分。

3.2　密钥预分配阶段

密钥预分配阶段是在节点部署到指定的区域之前,由基站或者可信的第三方来实施运行的。所有的节点都要预先分配一些信息,用于网络部署后节点间共享密钥的建立。

对于网络中的任意节点,本文采用分组[11]的方法代表节点标志符,对于每个组分配惟一的组标志符 CID,组内各成员节点再分配节点标志符 NID。对任意的节点 u,它拥有惟一的网络标志符 $ID_u=CID_u\parallel NID_u$。在节点部署之前,基站首先选择一对初始参量$\langle x_0=x_s,\lambda_0=\lambda_s\rangle$,并且按照式(2)和映射函数 $\phi(\cdot)$生成一个链,如下所示$\{\phi(x_1):id_1,\phi(x_2):id_2,\cdots,\phi(x_n):id_n\}$。其中 id_n 表示链的索引值,这里要求 n 的值要足够大。对于任意的节点 u,基站为其分配共享多项式 $f(u,y)=f(CID_u\parallel NID_u,y)$、节点的标志 ID_u、初始变量$\langle x_0=x_s,\lambda_0=\lambda_s\rangle$、混沌序列的索引 id_u、映射函数 $\phi(f(x_0,\lambda,n))$。其中 id_u 的分配按照以下算法[12]:对于任意节点,预分配时首先判断 $\phi(x_u)$ 的总数 $sum(\phi(x_u))$是否小于 t,如果 $sum(\phi(x_u))$小于 t,系统就将索引值分配给节点;反之,就取不同的 id_u 判断,直到满足条件为止。这样,经过密钥预分配之后,网络中所有节点都预分配了$(t+5)$个密钥信息,并且网络中拥有相同 id_u 的节点不会超过 t 个。

3.3　共享密钥建立阶段

本文假设节点能够根据协议选择簇头,如 ACE、FLOC[13],在簇形成之后,如果簇头被捕获或者能量耗尽,则根据算法在簇中剩余的节点中重新选取簇头。在该方案中,假定传感器节点的总数为 n 个,1 个基站节点,所有的簇头节点都可以与普通节点建立通信,普通节点间也可以直接建立通信。

这里以节点 u 为例,在这个阶段,节点 u 向它的邻居节点 v 广播自己的标志 ID_u 和混沌序列的索引值 id_u。邻居节点 v 收到广播后,首先判断 id_u 和 id_v 的关系,若 $id_u\geqslant id_v$,节点 v 则根据公式 $\phi(f(x_0,\lambda,n))$计算混沌序列在 id_u 的值 $\phi(x_{id_u})$,然后计算两者的共享密钥 $k_{vu}=\phi(x_{id_u})f(v,u)$;若 $id_u\leqslant id_v$,节点 v 则根据公式 $\phi(f(x_0,\lambda,n))$计算混沌序列在 id_v 的值 $\phi(x_{id_v})$,然后计算两者的共享密钥 $k_{vu}=\phi(x_{id_v})f(v,u)$。类似地,节点 u 根据相同的方式可以建立两者的共享密钥 $k_{uv}=\phi(x_{id_u})f(u,v)$或 $k_{uv}=\phi(x_{id_v})f(u,v)$,这样节点 u 和节点 v 可以建立安全的共享密钥。

节点 u 与所有的邻居节点建立密钥对之后,需要立即更新自己的多项式 $f'(u,y)=\phi(x_{id_u})f(u,y)$,然后节点 u 删除旧的多项式 $f(u,y)$、初始变量$\langle x_0=x_s,\lambda_0=\lambda_s\rangle$、映射函数 $\phi(f(x_0,\lambda,n))$,只保留更新后的多项式 $f'(u,y)$、节点的惟一标志 ID_u 以及混沌序列的索引值 id_u。这样,即使网络中的节点被捕获,也可以保证网络的安全性。

3.4　密钥的动态管理

无线传感器网络中的节点不可避免地会发生故障、能量耗尽或被攻击者捕获的情况,因此,为延长网络的生命周期,无线传感器网络中常常会有新节点加入,以替代损坏、被撤销的节

点继续工作。实施密钥的动态管理,能够使网络的安全性得到很大的提高。

1)节点的加入 本文假设增加的节点为 u,那么在部署之前,基站为其分配密钥信息:共享多项式 $f(u,y)=f(\mathrm{CID}_u \| \mathrm{NID}_u, y)$、节点的标志 ID_u、初始变量$\langle x_0=x_s, \lambda_0=\lambda_s\rangle$、混沌序列的索引 id_u、映射函数 $\phi(f(x_0,\lambda,n))$。其中 id_u 的分配遵循 3.2 节中的算法。节点 u 部署之后,向其邻居节点广播自己的信息——标志 ID_u 和混沌序列的索引值 id_u。在收到节点 u 的信息后,邻居节点 v 计算两者的共享密钥 $k_{vu}=f'(v,u)$,然后节点 v 将自己的标志 ID_v 和索引值 id_v 传送给 u。节点 u 在收到 v 的信息后,根据公式 $\phi(f(x_0,\lambda,n))$ 计算混沌序列在索引 id_v 的值 $\phi(x_{id_v})$,这里需保证 $id_u \leqslant id_v$,然后计算两者的共享密钥 $k_{uv}=\phi(x_{id_v})f(u,v)$。在与所有邻居节点建立共享密钥之后,节点 u 也需要更新自己的多项式信息并删除某些信息。

2)节点的删除 当普通传感器节点受损或能量耗尽时(假设该节点的标志为 ID_u,并且节点的这些不安全性都能探测到),基站广播该节点是不安全的,网络中的节点检测自己是否存储 ID_u 的密钥信息,如果有就将其密钥信息删除,节点 u 就不能与其他节点建立通信。这样,该节点就被排除到网络之外,不会威胁到网络的安全。

4 性能分析

4.1 节点抗捕获能力

无线传感器网络中节点的受损是不可避免的,因此为了保证网络的安全性,要求密钥管理方案具有较好的抗毁性能,即当部分节点受损后,尽可能少地暴露或者不暴露其他未受损节点的密钥信息[14]。在本方案中,假设每一个节点在部署后很短的时间内不会受损,这是因为建立密钥对所花费的时间比捕获节点的时间要短得多。节点中存储的密钥信息是在多项式的基础上配合混沌序列,两者共同产生密钥信息。在密钥建立阶段,虽然节点间的共享多项式是相同的,但是含有相同混沌序列索引的节点数不超过 t 个,也就相当于不超过 t 个节点含有相同的多项式,因此不管多少节点受损,也不会泄露其他节点的密钥信息。图 1 显示了不同方案中受损节点数和正常节点泄露密钥比例的关系。从图中可以看出,本方案在节点受损时不会泄露其他节点的密钥,能完全保证网络的安全。

4.2 通信和计算开销

文献[15]中指出,在无线传感器网络中节点的绝大部分能量消耗在数据传输上,其中 20% 消耗在共享密钥发现和建立的过程中,并且每传输 1 bit 数据所耗费的能量要比安全算法(节点 ID 的比较、hash 函数或多项式函数的计算)中计算 1 bit 数据高 2、3 个数量级,因此减少信息传输过程中的通信量将大大延长网络的寿命。在本方案中的共享密钥建立阶段,节点只需广播自己的标志和混沌序列的索引值,大大降低了网络的通信量,并且在密钥建立之后,所有的节点都能够与其通信范围内的节点建立安全通信,节点间信息的传输都是通过单跳实现的,因此所有节点都可以直接建立通信,而不需要中间节点如簇头转发数据,大大降低了通信开销。

在计算开销方面,多项式的模乘运算是一种开销较大的运算,但是这种方式有以下优点:a)与同样使用模乘运算的非对称加密算法(如 RSA、ECC)相比,它的计算量还是比较小的,并且在文献[5]中提出了减少计算量的方法;b)在密钥建立阶段,节点间只进行一次多项式的运算,在以后的通信过程中无须重新建立密钥。

4.3 节点存储量分析

传感器节点由于受体积和功耗的限制,其程序空间和内存空间比普通的计算机要弱很多,在密钥管理方案中必须考虑节点的存储量。下面进行详细分析。

在 RS 密钥分配方案中,每一个节点需要存储 s' 个 t 阶多项式,因此,节点的存储量为 $s'(t+1)\log q$,也就是每个节点需要存储 $s'(t+1)$ 个密钥(其中 s' 为多项式的个数,t 为多项式的阶数),网络支持的最大节点数为 $N=s\times(t+1)/s'$;在基于网格的密钥分配方案中,每个节点需要存储 $2m$ 个密钥其中 $m=\lceil\sqrt{N}\rceil=t+1$;而在本文提出的方案中,普通节点只需存储一个 t 阶多项式和混沌序列的索引值 id_u,因此需要存储$(t+2)$个密钥信息。

图 2 显示了不同方案中节点密钥存储量与网络规模的关系。从图 2 中可以看出,不管网络规模多大,本方案中节点分配的密钥信息都是固定且最低的,因此本方案具有很大的优势。

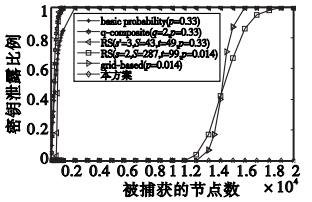

图1 受损节点数与正常节点泄露密钥比例

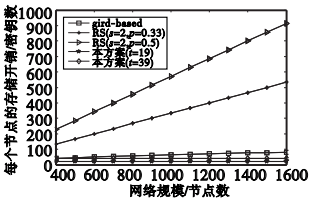

图2 节点密钥存储量网络规模

5 结束语

本文提出的密钥预分配方案将多项式与混沌序列有机地结合起来,在节点抗捕获能力方面采用多项式和混沌序列提高网络的安全性;在通信和计算开销方面节点间的通信都是通过单跳来实现的,不需要通过中间节点的转发,这大大减少了传输过程中的通信量;另外,节点在存储空间占有量上是很小的。同时,还有一些问题需要研究:对一些移动性较强的节点缺乏有效管理,这将是下一步的研究方向。

参考文献:

[1] AKYILDIZ I F, SU W, SANAKARASVBRAMANIAM Y, *et al*. Wireless sensor networks: a survey [J]. Computer Networks, 2002, 38(4): 393-422.

[2] 孙利民,李建中,陈渝,等. 无线传感器网络[M]. 北京:清华大学出版社,2005.

[3] ESCHENAUER L, GLIGOR V. A key-management scheme for distributed sensor networks[C]//Proc of ACM Conference on Computer and Communications Security. New York: ACM Press, 2002: 41-47.

[4] CHAN Hao-wen, PERRIG A, SONG D. Random key predistribution schemes for sensor networks[C]//Proc of IEEE Computer Symposium on Security and Privacy. Washingtom DC: IEEE Computer Society, 2003: 197-213.

[5] LIU Dong-gang, NING Peng. Establishing pairwise keys in distributed sensor networks[C]// Proc of ACM Conference on Computer and Communications Security. New York: ACM Press, 2003: 52-61.

(下转第 2607 页)

当网络爬虫模块存储新的 URL 到 URL 数据库的速度慢于 SQL 注入漏洞检测模块从 URL 数据库中取出数据进行检测的速度时,SQL 注入漏洞检测模块将面临没有新的 URL 可以获取的情况,该模块应等待网络爬虫模块将新的 URL 存储到 URL 数据库,以此来保证同步。

4　实验

根据上述的思想实现了以网络爬虫为基础的 SQL 注入漏洞检测程序,对网站 www. ***. com 和 bbs. ****. com(安全原因未给出网站的具体地址)进行检测,检测结果如表 1 所示。

表 1　检测结果

序号	存在 SQL 注入漏洞的 URLs	类型	方法	数据库
0	http://www. ***. com/index. php? gid = 55	字符型	GET	Oracle
1	http://bbs. ***. com/wap/index. php? gid = 55	字符型	GET	Oracle
2	http://bbs. ***. com/topicadmin. php?Action = moderate&fid = 2&infloat = yes&nopost = yes	字符型	POST	MS SQL
3	http://bbs. ****. com/indexfdc. php?gid = 3000&code = 2008&codeID = &radio = radio2&radio3 = radio	字符型	GET	Oracle
4	http://bbs. ****. com/search. php?formhash = 1a0611e0&srchtxt = 2008&radio = radio&radio1 = radio	字符型	POST	MS SQL

从表中可以看到,第一个网站检测到三个 SQL 注入漏洞,第二个网站有两个 SQL 注入漏洞,证明了将网络爬虫技术运用到 SQL 注入漏洞的检测是可行的。同时,由于网路爬虫技术的使用,网站的爬行深度增加了,网页覆盖面更广泛,被检测网页的数量更多,从而降低了漏报率。

5　结束语

本文改进了传统的网络爬虫技术,应用到对网站的 SQL 注入漏洞的检测中,降低了检测的漏报率,拓展了 SQL 注入漏洞的检测手段。同时,由于网络爬虫对系统资源和网络稳定性要求较高,今后研究工作的重点可以是在提高网络爬虫性能的同时降低系统资源和网络资源的占用率,并丰富和完善 SQL 注入漏洞检测的手段。

参考文献:

[1] National Vulnerability Database. National vulnerability database (NVD) CVE statistics[EB/OL]. (2009-12). http://web. nvd. nist. gov/view/vuln/statistics-results? cid = 4.

[2] OWASP. Top 10 2007[EB/OL]. (2009-11). http://www. owasp. org/index. php/Top_10_2007.

[3] 刘合叶. 多功能 SQL 注入检测系统的实现及攻击防范方法研究[D]. 北京:北京交通大学,2009.

[4] 陈小兵,张汉煜,骆力明,等. SQL 注入攻击及其防范检测技术研究[J]. 计算机工程与应用,2007,43(11):151-152.

[5] 赵亭,陆余良,刘金红,等. 基于表单爬虫的 Web 漏洞探测[J]. 计算机工程,2008,34(9):186-188.

[6] BANDHAKAVI S, BISHT P, MADHUSUDAN P, *et al*. CANDID: preventing SQL injection attacks using dynamic candidate evaluations[C]//Proc of the 14th ACM Conference on Computer and Communications Secirity. New York: ACM Press, 2007: 12-24.

[7] 周德懋,李舟军. 高性能网络爬虫:研究综述[J]. 计算机科学,2009,36(8):26-29.

[8] KUROSE J F, ROSS K W. 计算机网络——自顶向下方法与 Internet 特色[M]. 申震杰,王金伦,杜江,等译. 北京:机械工业出版社,2005:110-111.

[9] FRIEDL J E F. Mastering regular expressions[M]. 2nd ed. [S. l.]: O'Reilly Media Inc, 2005: 10-21.

[10] PESSOA J. Detecting SQL injection vulnerabilities in Web services[C]//Proc of the 4th Latin-American Symposium on Dependable Computing. Joao Pessoa, Brazil: IEEE Computer Society, 2009: 17-24.

[11] KEMALIS K, TZOURAMANIS T. SQL-IDS: a specification-based approach for SQL-injection detection[C]//Proc of ACM Symposium on Applied Computing. New York: ACM Press, 2008: 2153-2158.

[12] CHAPELA V. Advanced SQL injection[EB/OL]. [2005-11]. http://www. owasp. org/images/7/74/Adcance_SQL_Injection. ppt. .

[13] ANLEY C. (more) Advanced SQL injection[EB/OL]. [2002-06-18]. http://www. ngssoftware. com/papers/more_advanced_sql_injection. pdf.

[14] 沈寿忠. 基于网络爬虫的 SQL 注入与 XSS 漏洞挖掘[D]. 西安:西安电子科技大学,2009.

[15] 汤子瀛,哲凤屏. 计算机操作系统[M]. 西安:西安电子科技大学出版社,2004:46-48.

(上接第 2604 页)

[6] DU Wen-liang, DENG Jing, HAN Y S, *et al*. A pairwise key predistribution scheme for wireless sensor networks[J]. ACM Trans on Information and System Security, 2005, 8(2): 228-258.

[7] LIU Dong-gang, NING Peng, LI Rong-fang. Establishing pairwise keys in distributed sensor networks[J]. ACM Trans on Information and System Security, 2005, 8(1): 41-77.

[8] TRAYNOR P, CHOI H, CAO Guo-hong, *et al*. Establishing pairwise keys in heterogeneous sensor networks[C]//Proc of the 25th IEEE Conference on Computer Communications. Barcelona: IEEE Press, 2006: 1-12.

[9] BLUNDO C, DE S A, HERZBERG A, *et al*. Perfectly secure key distribution for dynamic conferences[J]. Information and Computation, 1998, 146(1): 1-23.

[10] 张化光,王智良,黄玮. 混沌系统的控制理论[M]. 沈阳:东北大学出版社,2003.

[11] 肖德贵,杨金,罗娟. 基于多项式和分组的无线传感器网络密钥管理方案[J]. 计算机应用研究,2009,26(2):680-682.

[12] XU Li, SHEN Jin-bo. A novel key pre-distribution scheme using one-way hash chain and bivariate polynomial for wireless sensor networks[C]//Proc of the 3rd International Conference on Anti-Counterfeiting, Security, and Identification in Communication. Piscataway NJ: IEEE Press, 2009: 575-580.

[13] ABBASI A A, YOUNIS M. A survey on clustering algorithms for wireless sensor networks[J]. Computer Communications, 2007, 30(14-15): 2826-2841.

[14] 苏忠,林闯,封富君,等. 无线传感器网络密钥管理的方案和协议[J]. 软件学报,2007,18(5):1218-1231.

[15] PERRIG A, SZEWCZYK R, TYGAR J, *et al*. SPINS: security protocols for sensor networks[J]. Wireless Networks, 2002, 8(5): 521-534.